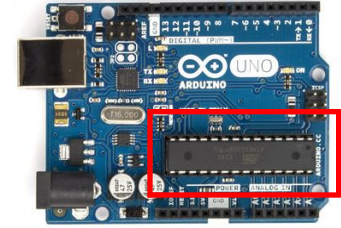


# Тема 1

## Электронные КОМПОНЕНТЫ

**Микроконтроллер** – это микросхема, в которой есть  
процессор,  
оперативная память,  
флеш-память,  
периферийные устройства.



Микроконтроллер **нужно программировать**, т.е. задавать ему логику поведения. (C или C++)

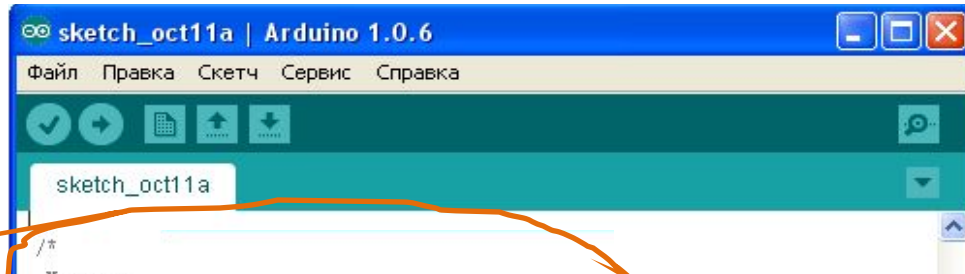
На микроконтроллер *приходят все сигналы*, поступающие на плату.

Микроконтроллер *раздаёт команды* всем устройствам, подключённым к плате.

**Области использования** микроконтроллеров:

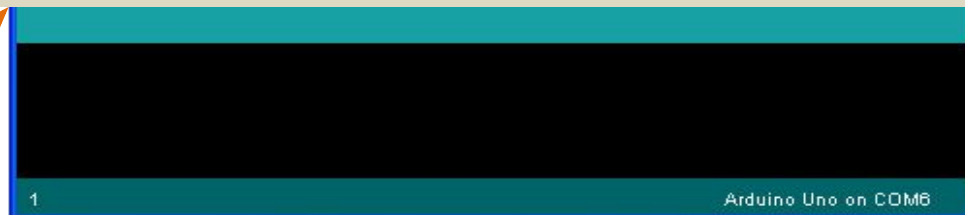
бытовые приборы,  
домофоны,  
радиоуправляемые игрушки,  
системы безопасности в автомобиле,  
роботы на конвейерах заводов,  
элементы управления летательными аппаратами т.п.

**Среда разработки** нужна для удобства программирования.



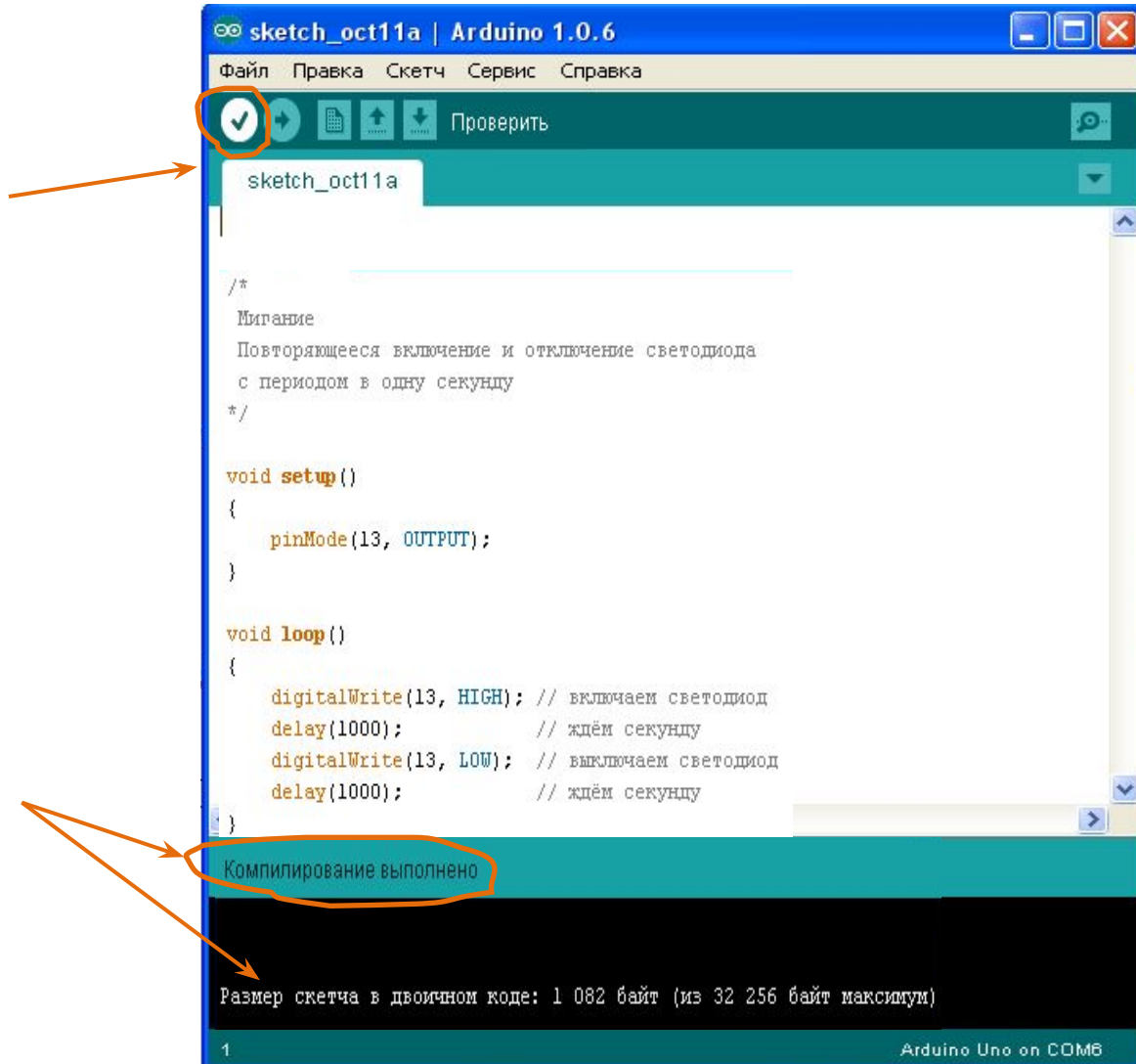
программный  
(скетч)

*Программный код* — это просто текст, написанный на специальном языке, который понятен и человеку и компьютеру.



окно среды разработки

**Компиляция** – это перевод программы, составленной на языке высокого уровня, в эквивалентную программу на низкоуровневом языке, близком машинному коду, выполняемый компилятором.



Программу нельзя читать и писать *как книгу*.

Любая программа *состоит из отдельных блоков*.

Начало блока кода в C/C++ обозначается левой фигурной скобкой **{**, его конец – правой фигурной скобкой **}**.

Блоки бывают *разных видов*.

Блоки в программе можно *менять местами*.

# Процедуры **setup** и **loop**

*установка, настройка*

*цикл, петля*

*пустота  
(процедура ничего  
не возвращает)*

```
void setup ()  
{  
}
```

*название  
процедуры*

```
void loop ()  
{  
}
```

```
/*  
*/
```

любое количество строк

*комментарий* Комментарии никак **не влияют** на работу программы, компилятор их не проверяет.

Комментарии используются для собственных пометок или **пояснений**.

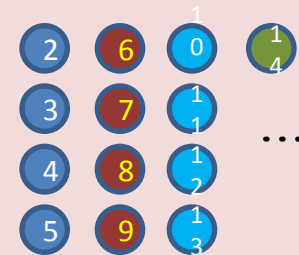
## процедура **setup**

```
void setup()
{
  pinMode(13, OUTPUT); ①
}
```

Каждая команда  
процедуры **setup**  
выполняется **один**  
**раз**

## процедура **loop**

```
void loop()
{
  digitalWrite(13, HIGH); // включаем светодиод
  delay(1000);             // ждём секунду
  digitalWrite(13, LOW);  // выключаем светодиод
  delay(1000);            // ждём секунду
}
```



Выполнение всех команд  
процедуры **loop** постоянно  
**повторяется (цикл)**



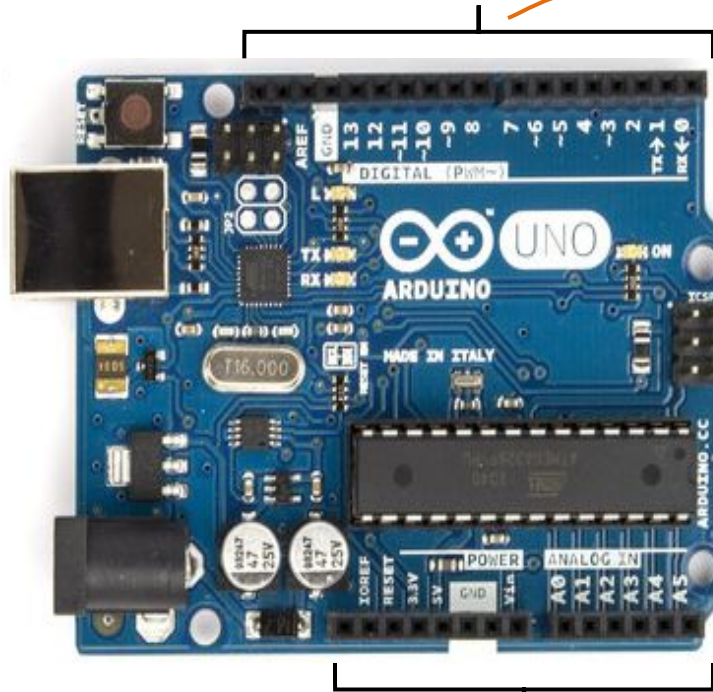
# Пронумерованные контакты платы называются **пинами**

К каждому контакту можно *подсоединять* какое-нибудь устройство

Пины Arduino могут работать и как *выходы* и как *входы*

Когда требуется чем-то **управлять**, то есть выдавать сигнал на устройство, нужно перевести управляющий пин в состояние работы **на выход**.

Когда сигнал **принимается** с устройства (прибора), нужно перевести управляющий пин в состояние работы **на вход**.



процедура **setup**

```
void setup()  
{  
  pinMode(13, OUTPUT);  
}
```

встроенная процедура

**pinMode(13, OUTPUT);**

имя процедуры

аргументы процедуры

**Аргумент** – это то, что передаётся в процедуру.

Аргументы друг от друга отделяются **запятой**.

Сколько у процедур аргументов зависит от назначения процедуры  
**(0, 1, 2, ..., 10, ...)**

процедура **setup**

**Номер пина**, с которым  
будем работать

```
void setup()
{
  pinMode(13, OUTPUT);
}
```

Устанавливает заданный  
**пин** в заданный **режим**:  
ВХОД или ВЫХОД

**Режим**, в который  
устанавливаем пин  
(выход, т.е. будет передаваться  
сигнал на светодиод)

процедура **loop**

```
void loop()
{
  digitalWrite(13, HIGH); // включаем светодиод
  delay(1000); // ждём секунду
  digitalWrite(13, LOW); // выключаем светодиод
  delay(1000); // ждём секунду
}
```

встроенная процедура

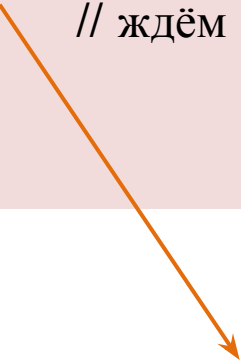
**Подает напряжение** на заданный пин

**HIGH** – это *встроенная константа (5 вольт)*

Светодиод должен **загореться**

## процедура **loop**

```
void loop()
{
  digitalWrite(13, HIGH); // включаем светодиод
  delay(1000);           // ждём секунду
  digitalWrite(13, LOW); // выключаем светодиод
  delay(1000);           // ждём секунду
}
```



**LOW** – это *встроенная константа (0 вольт)*

Светодиод должен **выключиться**

## процедура **loop**

```
void loop()
{
  digitalWrite(13, HIGH); // включаем светодиод
  delay(1000);           // ждём секунду
  digitalWrite(13, LOW); // выключаем светодиод
  delay(1000);           // ждём секунду
}
```

встроенная процедура

### Процедура **задержки**

Процессор перестаёт  
выполнять команды на  
некоторое время

**1000** — это *значение задержки* в  
миллисекундах

1000 мс = 1 с

## Типы данных

**Объявление (инициализация) переменных**

```
int ledPin = 13;  
int dotDelay = 200;
```



***Переменная*** — это некоторое **имя**, под которым скрывается какое-либо число или строка.

Имя переменной мы придумываем сами. Оно может состоять из латинских символов и цифр, но не может начинаться с цифры.

# Типы данных

*Тип* переменной

```
int ledPin = 13;  
int dotDelay = 200;
```

Имя переменной

Название	Тип	Размер в байтах	Диапазон значений
целый	<b>int</b>	2	-32768..32767
логический	<b>boolean</b>	1	<b>true</b> или <b>false</b>
символьный	<b>char</b>	1	-128 или 127
длинный целый	<b>long</b>	4	-2 147 483 648..2 147 483 647



## Встроенные константы

***HIGH*** – логическая единица,  
5 вольт

***LOW*** – логический ноль,  
0 вольт

***OUTPUT*** – режим выхода

***INPUT*** – режим входа

константы:

***INPUT, OUTPUT, LOW, HIGH,***

пишутся заглавными буквами, иначе  
компилятор их не распознает и выдаст ошибку.

## Встроенные процедуры

`pinMode(x, y)` - устанавливает **заданный пин** в **заданный режим**: ВХОД ИЛИ ВЫХОД  
(режим пина)

`digitalWrite(x, y)` - подает **необходимое напряжение** на **заданный пин**  
(цифровая запись)

`delay(x)` - останавливает дальнейшее выполнение программы на **заданное количество миллисекунд**  
(задержка)

**Электричество** – совокупность явлений, обусловленных существованием, взаимодействием и движением электрических зарядов.

**Электрический заряд (количество электричества)** – это физическая величина, определяющая способность тел быть источником электромагнитных полей и принимать участие в электромагнитном взаимодействии.

Электрический заряд измеряется в **кулонах**.

**1 кулон** – это заряд тел, которые на расстоянии в **1 метр** притягиваются друг к другу с силой **1 ньютон**.

**Ток** – направленное движение заряженных частиц.

**Сила тока** – это величина, равная отношению **количества заряда**, проходящего через поперечное сечение проводника, **к времени** его прохождения.

Сила тока измеряется в **амперах**.

# Как увеличить силу тока?

- 1) провод увеличить в диаметре;
- 2) выбрать материал с меньшим удельным сопротивлением;
- 3) увеличить «напор» электронов, т.е. поставить насос.

напряжение

источник питания

напряжение измеряется в **вольтах**

у источника питания есть **два полюса**

Источник питания даёт **1 вольт** напряжения, если при перемещении **1 кулона** заряда между его полюсами совершается работа в **1 джоуль**.

**Ток течёт от плюса к минусу**

**+**

**-**

**Конденсатор** – это устройство для **накопления** электрического заряда и энергии электрического поля.

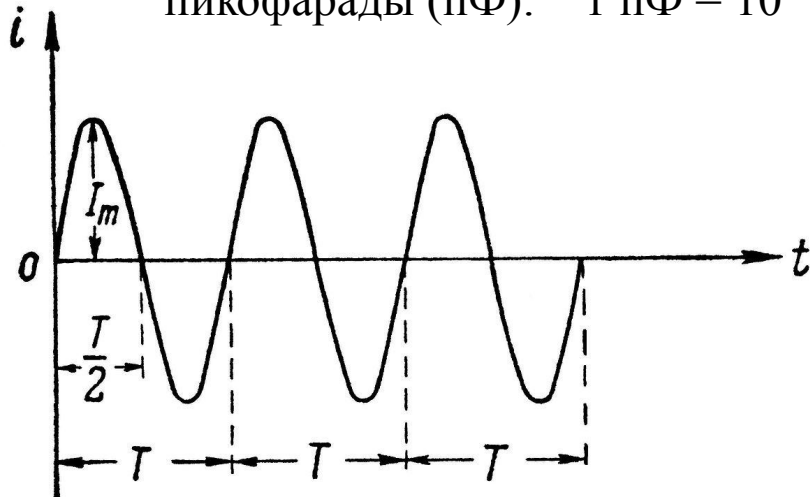
Конденсатор используется для **сглаживания** пульсации **напряжения**.

Ёмкость конденсатора измеряется в **фарадах (Ф)**

микрофарады (мкФ)  $1 \text{ мкФ} = 10^{-6} \text{ Ф}$

нанофарады (нФ)  $1 \text{ нФ} = 10^{-9} \text{ Ф}$

пикофарады (пФ):  $1 \text{ пФ} = 10^{-12} \text{ Ф}$



Развернутая диаграмма периодического переменного тока

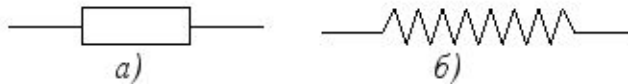
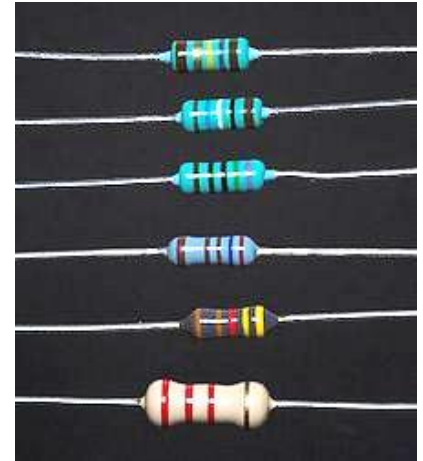


**Переменный ток**, в отличие от тока постоянного, непрерывно изменяется как по величине, так и по направлению, причем изменения эти происходят периодически, т. е. точно повторяются через равные промежутки времени.

# Как укротить электричество?

**Резистор** – элемент электрической цепи, предназначенный для использования его электрического **сопротивления**.

**Резистор** – искусственное «препятствие» для тока. Резистор **ограничивает** силу тока, переводя часть электроэнергии в тепло.



- а) обозначение на схемах в России и Европе
- б) обозначение на схемах в США и Японии

Сопротивление измеряется в **омах**

## Характеристики резистора

Сопротивление	R	Ом
Точность (допуск)	±	%
Мощность	P	Ватт

### Типовые номиналы для экспериментов



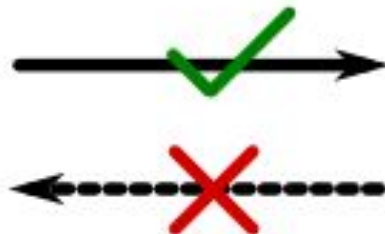
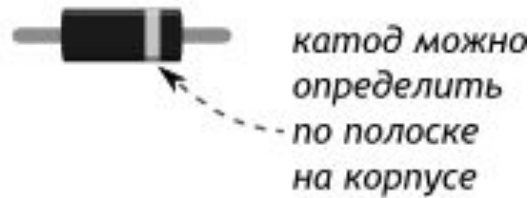
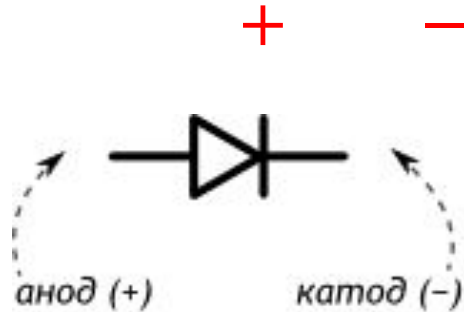
360 Ом  $\approx$  36  $\times$  10<sup>1</sup> ±5%

основание 2, 3 или 4 полосы      множитель предпоследняя полоса      точность последняя полоса

чёрный	0	10 <sup>0</sup>	
коричневый	1	10 <sup>1</sup>	±1%
красный	2	10 <sup>2</sup>	±2%
оранжевый	3	10 <sup>3</sup>	
жёлтый	4	10 <sup>4</sup>	
зелёный	5	10 <sup>5</sup>	±0,5%
синий	6	10 <sup>6</sup>	±0,25%
фиолетовый	7	10 <sup>7</sup>	±0,1%
серый	8	10 <sup>8</sup>	±0,05%
белый	9	10 <sup>9</sup>	
золото		10 <sup>-1</sup>	±5%
серебро		10 <sup>-2</sup>	±10%

# Как укротить электричество?

**Диод** – это клапан, пропускающий ток только в одном направлении, от **анода** к **катоде**.





**Светодиод** – это вид диода, который светится, когда через него проходит ток (токовый прибор).

Светодиоды подключают вместе с **токоограничивающим резистором** (балластным)



Для питания 1 светодиода  
на 20 мА от 5 В  
используйте резистор  
от 150 до 360 Ом.

пода  
рчка

# Макетная плата

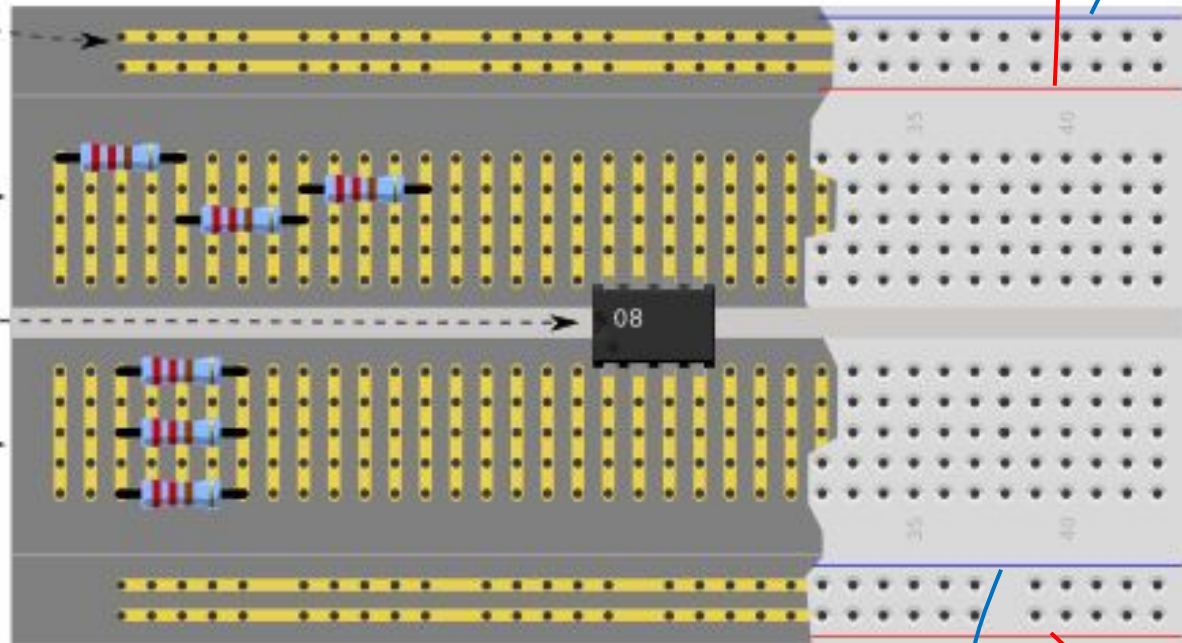
Длинные рельсы по бокам обычно используют для соединения с источником питания

напряжение (5 Вольт)

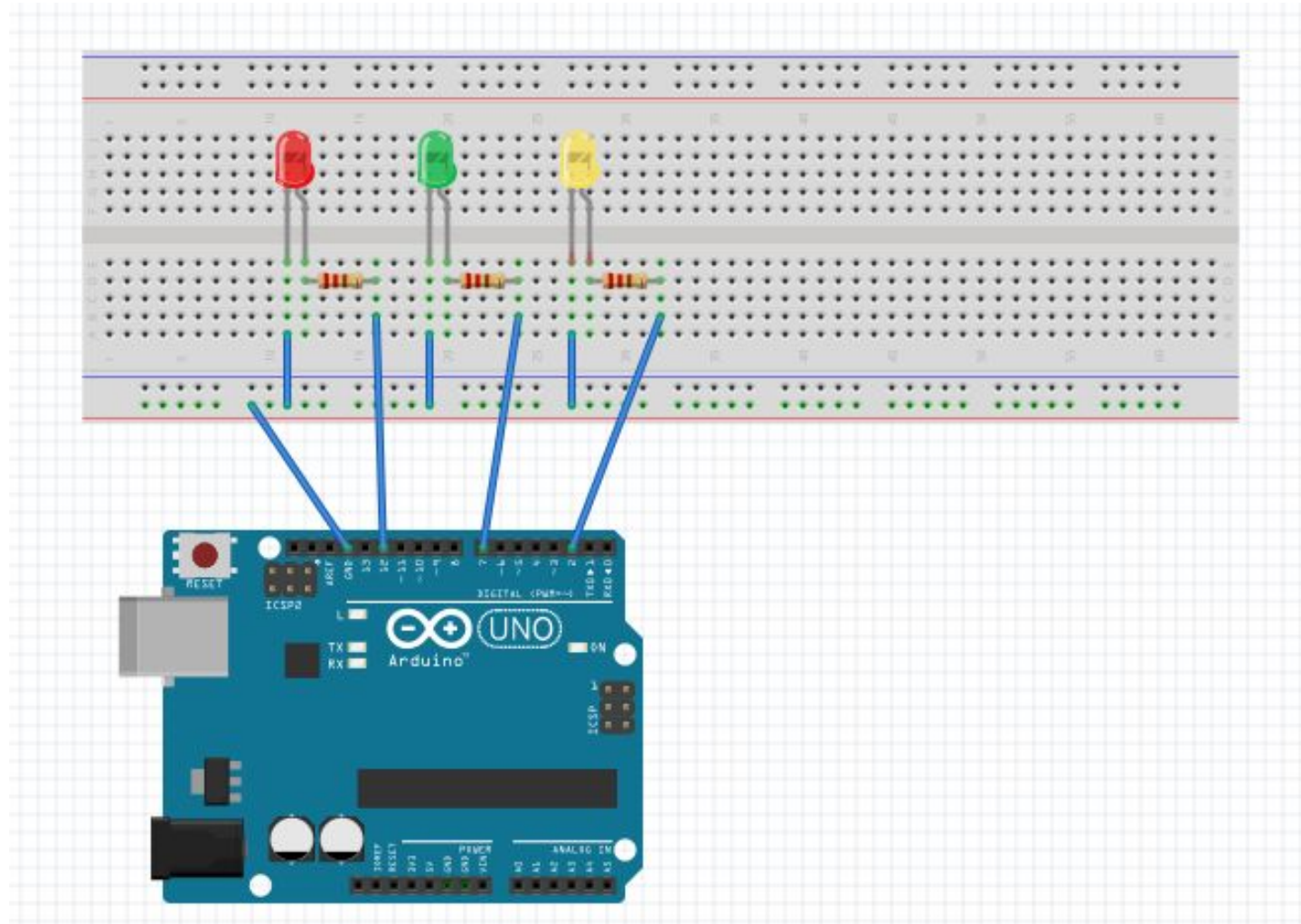
Пример последовательного соединения компонентов

Разрыв посередине используют для компонентов с двумя рядами ног

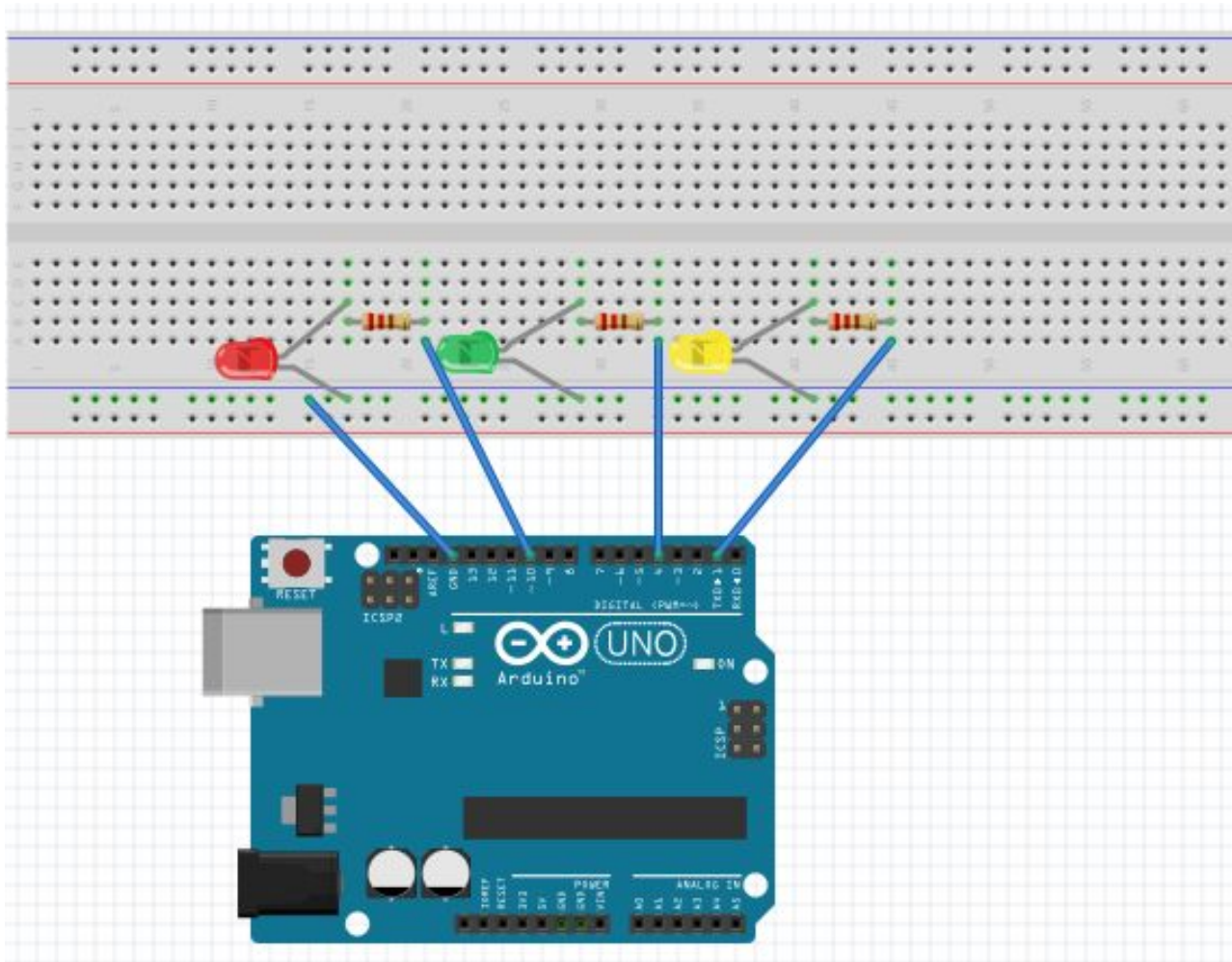
Пример параллельного соединения компонентов



# Пример сборки схемы на макетной плате



# Пример сборки схемы на макетной плате



## Пример сборки схемы на макетной плате

